REMARKS

I. Office Action Summary

In the Office Action dated February 9, 2005, the Examiner rejected claims 1-9 and 12-23. The Examiner requested Applicant's assistance in understanding the reason behind the citation of prior art references. Claims 1-4 and 17-18 were rejected under 35 USC §101 as directed to non-statutory subject matter. Claims 1-9 and 12-23 were rejected as indefinite under 35 U.S.C. §112, second paragraph. Finally, all of the pending claims were rejected as anticipated by each of the following:

Reference	Statutory Basis
Beitz et al. ("Service Location In An Open Distributed Environment" Second International Workshop on Services in Distributed and Networked Environments, June 1995)	§102(b)
Perkowski (U.S. 5,950,173)	§102(e)
Hudetz et al. (U.S. 5,978,773)	§102(e)
Call (U.S. 5,913,210)	§102(e)

II. Request For Information Regarding the IDS

Applicant has cited the references currently of record in response to the duty of disclosure under 37 CFR § 1.56 and in the format required by MPEP § 609. The cited references include relevant references cited in corresponding foreign applications and references in the possession of Applicant. Applicant is reluctant to go beyond the duty of disclosure mandated by the rules and describe the teachings of each reference out of concern that any such characterization would be misconstrued as erroneous or misleading. The undersigned has discussed the pertinence of the cited art with the Examiner and believes that the Examiner's questions have now been addressed.

III. Rejections Under 35 U.S.C. § 101

Applicant respectfully disagrees that claims 1-4 and 17-18 were directed to non-statutory subject matter. Applicant has cancelled these claims to focus more directly on aspects of the invention recited in new claims 24-49. Accordingly, Applicant submits that this rejection is now moot. Furthermore, none of new claims 24-49 utilize the term "protocol" identified as objectionable by the Examiner.

IV. Rejections Under 35 U.S.C. § 112, Second Paragraph

Applicant respectfully disagrees that claims 1-9 and 12-23 were indefinite. Applicant has cancelled these claims to focus more directly on aspects of the invention recited in new claims 24-49. Accordingly, Applicant submits that this rejection is now moot. New claims 24-49 do utilize the terms "redirect message" and "hash function" in several instances; however, Applicant submits that these terms are properly understandable in the context of the new claims in which they are now being used.

V. Rejections Under 35 U.S.C. § 102

Applicant respectfully disagrees with the Examiner's rejections of claims 1-9 and 12-23 as anticipated by the 4 cited references. However, in order to expedite allowance of claims 24-49 directed to certain aspects of the invention, claims 1-9 and 12-23 have been cancelled.

VI. New Claims 24-29

New claims 24-49 relate to systems and methods that decentralize the processing requirements that can burden traditional databases, such as relational databases. The claimed method and system generally relates to a highly scalable and dynamic way of handling extremely large amounts of data by distributing the way data location information is handled (i.e. the information on where a piece of data is stored in a network), distributing it across a logical pool of participating machines, and distributing data location computation across participating machines, for example by allowing the clients that request the location of data to participate in the computation.

One aspect of a system configured with a communication protocol to implement this data handling strategy is the redirect message supported by the communication protocol. Examples of one type of redirection include the "well-known" and "passed function" redirection described in the specification, for example at pages 24-26.

In 'well-known function' redirection both servers and clients compute the same function on the same data to generate the same location results. This guarantees that a system configured with the protocol described (NDTP) distributes computation across clients and servers. The disclosure illustrates this using the standard hashpjw function from a reference computer science text book.

> The first appropriate NDTP server is selected from the table in the NDTP_RDR_RSP message by applying a well-known function to the identifier string and using the function result as an index into the NDTP server table. The well-known function preferably applied is the *hashpjw* function presented by Aho, Sethi and Ullman in their text Compilers, Principles, Techniques and Tools (Specification at p. 24, ll. 26–31).

That is, the system distributes hash-function computation, or creates a "distributed hash table". This is fundamentally different than relational database management systems, which are not hash tables. This well-known function can be applied to both clients and servers in their original programming.

> The first variant of the NDTP_RDR_RSP function mechanism specifies a well-known function that all NDTP server and client implementations know when they are programmed, and the NDTP_RDR_RSP message carries a table of NDTP server URLs. (Specification at p.24, ll.20).

As discussed in the specification in one embodiment, when a client requests data from a server for which the server is not responsible, the server returns a NDTP_RDR_RSP message containing a table listing the currently deployed server configuration. This does not indicate where the data is, only which servers currently are operating in a server pool. NDTP_RDR_RSP returns the very same information that the servers use to compute where to put the data in the first place, so the client will compute the location just like the servers originally knew how to store it. Both servers and clients hash the location, using the same information. Receiving NDTP_RDR_RSP, a client runs the same computation, on the same data and therefore computes the same result.

> The appropriate NDTP server is selected from the table in the NDTP_RDR_RSP message by applying a well-known function to the identifier string and using the function result as an index into the NDTP server table (Specification at p. 24, ll. 26–28).

In another disclosed redirection approach, the 'communicated function' redirection, servers use NDTP_RDR_RSP message to pass new function definitions to clients, rather than requiring clients and servers to share the same "well-known" function.

> The second variant of the NDTP_RDR_RSP function mechanism specifies that a general function description will be sent to the NDTP client in the NDTP_RDR_RSP message. The client will apply this function to the identifier string and the output of the function will be the NDTP server URL to which to send NDTP requests for the particular identifier string. (Specification at p. 25, ll. 32–p. 26, ll. 2).

Communicated function allows passing entirely new functions to clients, and therefore allows modifying system behavior at the moment of interaction with the NDTP distributed system. Rather than recalling millions of clients to replace the well-known function in both clients and servers, this allows dynamically remapping the identifier string space to the deployed system in the field. An example of this might be if a company installs one or more NDTP (data location) servers and then learns that it would benefit by portioning certain "General Electric" identifier location strings to a new location. Using communicated function redirection via NDTP_RDR_RSP would allow this.

> The advantage of this technique over the well-known function approach is that it allows application-specific partitions of the identifier string space. This can permit useful administrative control. For example, if General Electric manages all identifies beginning with the prefix "GE", a general function can be used to make this selection appropriately. The disadvantage of using a general function is it may be less efficient to compute than a well-known function, (Specification at p. 25, ll. 32–p. 26, ll. 2).

Finally, communicated function redirection may allow changing the processing behavior of clients and servers dynamically that are already deployed in the field.

Each of new claims relates to aspects of the redirection process and system configuration. Claim 24, for example, relates to the process supported by redirection (including both the well-known function and the communicated function versions as recited in claims 26 and 28). Support for claim 24 may be found throughout the specification as discussed above and in FIGS. 10-12.

Referring to claim 24, when a user enters a request at a client for data at a first client, the data query includes an identifier string (e.g. a file name or URI) and expects to receive information (a location string) on where the data corresponding to that filename resides. The location strings are stored in one or more data location servers in the network. The first client then transmits a data location request to a server to retrieve the location string associated with the identification string in the data query. If the server is not a data location server, then that server can turn around and operate as a "next client" transmitting the data location request to a server logically associated with it. This process of the queried server then functioning as a client to ask another server in the logical chain repeats until the data location request is transmitted to a data location server. During this navigation process, if a data location server does not possess the sought after location string, it transmits a redirect message back along the communication path the data location request traveled to the first client. The redirect message that is sent back to the first server contains information with which the first client is configured to determine a location of a second data location server that does contain the sought after location string. In other words, the redirect message allows the first server to calculate a new data location server address so that the first server "redirects" its query to the right data location server. An example of a network topology that the method of claim 24 may work with is shown in FIG. 10. However, the method of claim 24 is not limited to any particular physical hierarchy, simply a logical hierarchy.

Similarly, the method of claim 33 also includes includes the feature of redirection, however claim 33 addresses a more simplified scenario such as shown in FIG. 12, where there are not necessarily any intervening devices between the client and the data location servers.

New independent claims 41 and 42 relate to system configurations adapted to incorporating software instructions implementing the redirection process.

The specific network deployments of FIGS. 10-13 are merely illustrative examples to aid discussion because diagramming the combinatorial relationships can reach into factorial complexity. Discussion relating to this is found in the specification:

> In FIG. 10, the client 112 and data repository 114 of FIG. 11 were merged into the single client entity 106 for ease of discussion. Their distinction can now be separated and identified in order to illustrate the storage and retrieval of data in a distributed data collection" (Specification at p. 27, ll. 30—p. 28, ll. 2).

FIG 11. illustrates a client (112) that queries the server pool (110) for location information. Receiving the location information, the client queries a "data repository" (114) in the distributed data collection. As an example of the dynamism of the system, FIG. 10 labels:

(a) 108 to represent a collection of a highest logical level of servers ("Level 1" servers)
(b) 104 to represent next logical level of servers ("Level "2" servers)
(c) 106 to represent a collections of clients and repositories, in which ANY client can connect to ANY other client acting like a repository, and *visa versa*.

This can become exponentially complex, theoretically, using the claimed redirection mechanisms:

> "NDTP supports two specific redirection mechanisms, which are not mutually exclusive and may be combined in any way within a single NDTP server constellation" (Specification at p. 29, ll. 5–7).

In a hypothetical system that needs to track 100s of billions of objects across 100s of millions of clients and repositories in which any object can "spontaneously and dynamically" change location from any client to any other client in real-time, there are significant challenges. The claimed redirection feature is intended to help address and solve problems like these, using a theoretically one-point-failure-retry redirection method that configurably distributes computation among clients and servers.

None of the previously cited references teach or suggest the features of claims 24-49.

Beitz et al. discloses a system that needs some type of improved data management system, for example see p. 32 of Beitz, but fails to address how that data management system would be implemented. Accordingly, Beitz does not teach or suggest a redirection mechanism as recited in the pending claims.

Call discloses mapping UPC codes to IP addresses in a standard relational database using mirrored servers each having a complete copy of a look-up table. (See Col. 3, line 61 – Col. 4, line 9). Call does not disclose or suggest the claimed redirection mechanism.

Perkowski discloses linking consumers to product information using a relational database. Perkowski teaches the use of traditional relational databases using centralized or mirrored deployment. (See Col. 7, lines 38-42; Col. 11, lines 34-61; Col. 12, lines 30-35; and Col. 18, lines 13-35). Perkowski fails to teach or suggest a redirect message containing information for a client or server to determine the location of another data location server, a client or server computing redirection information or of the location of servers changing dynamically.

Similarly, Hudetz et al. discloses mapping UPC numbers to IP addresses that can provide the consumer more information on the product. The disclosure mentions in passing a database that stores the UPC-to-IP data table and does not disclose or suggest a dynamic data management system with a redirection function. (See FIG. 4; Col. 7, lines 1-9).

VII. Conclusion

Applicant wishes to thank Examiner Thompson for the courtesy extended in the in-person interview with Applicant and the undersigned on June 17, 2005. Applicant has cancelled claims 1-9 and 12-23, and added claims 24-49. Applicant submits that no new matter has been added. In light of the above remarks and amendments, Applicant submits that claims 24-49 are in condition for allowance. If any issues arise or questions remain, the Examiner is invited to contact the undersigned at the number listed below in order to expedite disposition of this case.

Respectfully submitted,

Kent E. Genin
Registration No. 37,834
Attorney for Applicant

BRINKS HOFER GILSON & LIONE
P.O. BOX 10395
CHICAGO, ILLINOIS 60610
(312) 321-4200